



TITLE:

# On fine structures between Church-style and Curry-style $\lambda$ -terms (Proof theory and complexity)

AUTHOR(S):

Fujita, Ken-etsu

---

CITATION:

Fujita, Ken-etsu. On fine structures between Church-style and Curry-style  $\lambda$ -terms (Proof theory and complexity). 数理解析研究所講究録 2013, 1832: 73-87

ISSUE DATE:

2013-04

URL:

<http://hdl.handle.net/2433/194853>

RIGHT:

# On fine structures between Church-style and Curry-style $\lambda$ 2-terms

Ken-etsu Fujita (Gunma University)

December 14, 2012

## Abstract

We introduce a class of 2nd-order  $\lambda$ -terms with fine structures between so called Church-style and Curry-style. Here,  $\lambda$ -terms in the style of Curry are considered as atomic, and we adopt four term-constructors: (i) Domains ( $D$ ) for  $\lambda$ -abstraction, (ii) Lambdas ( $\Lambda$ ) for type-abstraction, (iii) Holes ( $[]$ ) for type-application, and (iv) Types ( $[A]$ ) to be filled into a hole. Then applying the term-constructors to Curry-style provides the set of 12 styles of  $\lambda$ 2-terms in total, where Church-style can be regarded as a top and Curry-style is a bottom. We examine which term-constructor determines decidability of type-checking and type-inference problems of  $\lambda$ 2-terms. This study reveals fine boundaries between decidability and undecidability of the type-related problems.

## 1 Introduction

Second-order  $\lambda$ -terms in the style of Church consist of variables, applications,  $\lambda$ -abstractions, type applications and type-abstractions [2].

$$M ::= x \mid MM \mid \lambda x:A.M \mid M[A] \mid \Lambda X.M$$

On the other hand,  $\lambda$ -terms in the style of Curry is the same as those of type-free  $\lambda$ -calculus. As a natural combinatorial problem, we can consider  $\lambda$ -terms with fine structures between Curry-style and Church-style. From the viewpoint of components of  $\lambda$ -terms, we take (i) domains of  $\lambda$ -abstraction, (ii) type abstractions  $\Lambda$ , (iii) holes  $[]$  to be filled with a type, and (iv) type information (polymorphic instance) to be inserted into a hole, as primitive term constructors for fine structures. We write  $D$ ,  $\Lambda$ ,  $[]$ , and  $[A]$ , respectively, for the constructors. Then, based on the Curry-style, the following 12 styles (structures) for  $\lambda$ -terms can be defined as a combination of four constructors. We write ST for the set of 12 styles, as follows:

- Church-style [2] denoted by Ch has constructors ( $D, \Lambda, [A]$ )
- Domain-free style [4] denoted by Df has ( $\Lambda, [A]$ )
- Type-free style [7] denoted by Tf has ( $\Lambda, []$ )
- Hole-application style [8] denoted by Hole has ( $D, \Lambda, []$ )
- ( $D, [A]$ ), ( $D, []$ ), ( $D, \Lambda$ ), ( $\Lambda, [A]$ ), ( $\Lambda, []$ ), ( $D$ ), ( $\Lambda$ )
- Curry-style [2] denoted by Cu has ( $$ )

The fine structures between Curry-style and Church-style are presented in the following picture, see Figure 1. Upper arrows on the cubes denote adding domains of  $\lambda$ -abstraction, where we only depict one upper arrow among a total of 6 upper arrows in the picture. Four right arrows on the left cube

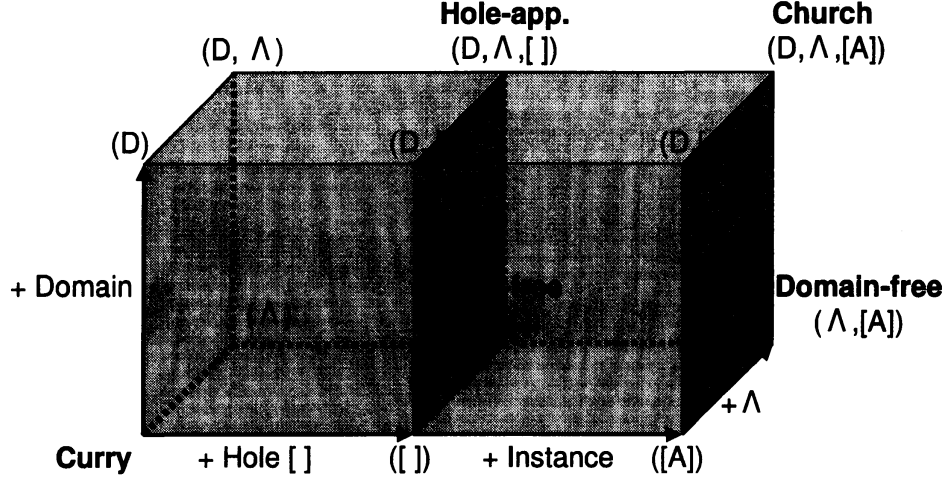


Figure 1:  $\lambda$ 2-terms with fine structures between Curry-style and Church-style

denote adding holes  $[ ]$ , other right arrows on the right cube denote adding polymorphic instance  $[A]$ , and six back arrows denote adding type abstractions  $\Lambda$ .

An order is defined on ST: Curry-style is the bottom, Church-style is the top, and  $s < t$  if we have an arrow from  $s$ -style to  $t$ -style for  $s, t \in \text{ST}$ .

The picture shows that terms on the upper plane contain domains of  $\lambda$ -abstraction, where the set of styles on the upper plane is denoted by UpP. On the other hand, the set of styles on the lower plane is denoted by LwP. Terms on the back plane contain type abstractions  $\Lambda$  where the set of styles on the back plane is denoted by BkP, and terms on the middle plane contain holes  $[ ]$  where the set of styles on the middle plane is denoted by MiP. Terms on the rightmost plane contain polymorphic instance  $[A]$ , where the set of styles on the right plane is denoted by RiP. The set of styles on the leftmost plane is denoted by LeP.

The first problem is how to define inference rules for each system. The second problem is how to define reduction rules for each system. For this, we call a system normal, if the system contains both  $\Lambda$  and either  $[ ]$  or  $[A]$ , or contains neither  $\Lambda$  nor  $[ ]$ . Namely, systems of Ch, Hole, Df, Tf, (D), and Cu are normal.

We study decision problems parametrized by  $\lambda$ -terms with an intermediate structure of the cubes, and investigate critical conditions for the decidability property from the viewpoint of the constructors  $(D, \Lambda, [ ]$ , and  $[A]$ ). In this paper, as decision problems we adopt the type checking (TCP), type inference (TIP), and typability (TP) problems for second-order  $\lambda$ -terms with fine structures. Then we examine what constructor determines essentially (un)decidability of the problems.

## 2 Preliminary

### Definition 1 (Type-related problems parameterized with styles)

1. *Type checking problem of  $s$ -style terms denoted by TCP( $s$ ) :*  
*Given an  $s$ -style  $\lambda$ -term  $M$ , a type  $A$ , and a context  $\Gamma$ , determine whether  $\Gamma \vdash_s M : A$ .*
2. *Type inference problem of  $s$ -style  $\lambda$ -terms denoted by TIP( $s$ ) :*  
*Given an  $s$ -style  $\lambda$ -term  $M$  and a context  $\Gamma$ , determine whether  $\Gamma \vdash_s M : A$  for some type  $A$ .*
3. *Typability problem of  $s$ -style terms denoted by TP( $s$ ) :*  
*Given an  $s$ -style  $\lambda$ -term  $M$ , determine whether  $\Gamma \vdash_s M : A$  for some context  $\Gamma$  and type  $A$ .*

Styles	TCP	TIP	TP
Church	yes $\hookrightarrow$	yes $\hookrightarrow$	no [10]
Hole-application	yes $\longleftrightarrow$	yes $\hookrightarrow$	no [8]
Domain-free	no $\longleftrightarrow$	no $\longleftrightarrow$	no [6]
Type-free	no $\longleftrightarrow$	no $\hookrightarrow$	no [7]
Curry	no $\longleftrightarrow$	no $\hookrightarrow$	no [11]

Table 1: Styles of  $\lambda 2$ -terms and decidability of TCP, TIP, TP**Proposition 1 (Reductions between type-related problems)**

1.  $\text{TCP}(s) \hookrightarrow \text{TIP}(s)$  for any  $s \in \text{ST}$ .
2.  $\text{TIP}(s) \hookrightarrow \text{TCP}(s)$  for any  $s \in \text{LwP} \cup \text{MiP} \cup \text{LeP}$ .
3.  $\text{TIP}(s) \hookrightarrow \text{TP}(s)$  for any  $s \in \text{UpP} \cup \{\text{Df}, ([A])\}$ .
4.  $\text{TP}(s) \hookrightarrow \text{TIP}(s)$  for  $s \in \text{LwP}$ .

*Proof.* 1.  $\Gamma \vdash_s M : A$  if and only if  $\Gamma, z : A \rightarrow Z \vdash_s zM : B$  for some  $B$ , where  $z, Z$  are fresh variables.

2. Let  $s \in \text{LwP}$ .  $\Gamma \vdash_s M : B$  for some  $B$  if and only if  $\Gamma, z : Z \vdash_s (\lambda v.z)M : Z$ , where  $z, v, Z$  are fresh variables with  $z \neq v$ .

Let  $s \in \text{MiP}$ .  $\Gamma \vdash_s M : B$  for some  $B$  if and only if  $\Gamma, z : \forall X.(X \rightarrow Z) \vdash_s z[M] : Z$ , where  $z, Z$  are fresh variables.

Let  $s \in \text{LeP}$ .  $\Gamma \vdash_s M : B$  for some  $B$  if and only if  $\Gamma, z : \forall X.(X \rightarrow Z) \vdash_s zM : Z$ , where  $z, Z$  are fresh variables.

3. Let  $s \in \text{UpP}$ . Let  $\Gamma = \{a_1 : A_1, \dots, a_n : A_n\}$  and  $z$  be a fresh variable.  $\Gamma \vdash_s M : B$  for some  $B$  if and only if  $\Sigma \vdash_s z(\lambda a_1 : A_1 \dots \lambda a_n : A_n.M) : B$  for some  $B$  and some  $\Sigma$ .

Let  $s \in \{\text{Df}, ([A])\}$ . Let  $\Gamma = \{a_1 : A_1, \dots, a_n : A_n\}$ .  $\Gamma \vdash_s M : B$  for some  $B$  if and only if  $\Sigma \vdash_s M_0 : B$  for some  $B$  and some  $\Sigma$ , where  $M_0 = z_0(z_1(z[\forall X.X]))(z_1z)(z[(A_1 \rightarrow \dots \rightarrow A_n \rightarrow Y) \rightarrow Y](\lambda a_1 \dots \lambda a_n.yM))$ , and  $z_0, z_1, z, y, Y$  are fresh variables.

If  $\Gamma \vdash_s M : B$  for some  $B$ , then  $M_0$  is typable. Because type of  $z$  is assigned to  $\forall X.X$ .

In turn, if  $M_0$  is typable then type of  $z$  should be a universal type, to say,  $\forall X.F(X)$ , where  $F$  is a second-order variable with arity 1. From consistent typability of the two occurrences of  $z_1$ , we have the following unification equation [5]:

$$F(\forall X.X) \doteq \forall X.F(X)$$

Observe that the only solution to the unification equation is  $[F := \lambda x.x]$ , i.e., the identity function, which implies that type of  $z$  is  $\forall X.X$ . Hence, we can recover the context  $\Gamma$ .

4. Let  $\{x_1, \dots, x_n\} = \text{FV}(M)$ .  $\Sigma \vdash_s M : B$  for some  $B$  and some  $\Sigma$  if and only if  $\vdash_s \lambda x_1 \dots \lambda x_n.M : B$  for some  $B$ .  $\square$

We summarize already known results on the problems for  $\lambda 2$ . Table 1 shows the decidability results and relations on the type-related problems. Here, “yes” means that a problem is decidable and “no” undecidable. TCP and TIP have the boundaries between hole-application and domain-free. Compared with Church-style, TIP remains decidable even after deleting polymorphic instance information on application of  $(\forall E)$ . However, on application of  $(\rightarrow I)$ , deleting polymorphic domains makes TIP undecidable. Therefore, polymorphic domains are considered as the most essential information for (un)decidable TIP. In this paper, we examine System (D), Curry-style with explicit domains.

### 3 System (D): Curry-style plus explicit domains

We introduce System (D) and study the type-inference and type-checking problems of the system.

- Types

$$\begin{aligned} X &\in \text{TypeVars} \\ A \in \text{Types} &::= X \mid (A \rightarrow A) \mid \forall X.A \end{aligned}$$

- Terms:

$$M \in \text{Terms} ::= x \mid (\lambda x : A.M) \mid (MM)$$

- Reduction rule:

$$(\lambda x : A.M)N \rightarrow_\beta M[x := N]$$

- Inference rules:

$$\begin{aligned} &\frac{}{\Gamma, x : A \vdash_D x : A} (\text{var}) \\ &\frac{\Gamma, x : A_1 \vdash_D M : A_2}{\Gamma \vdash_D \lambda x : A_1. M : A_1 \rightarrow A_2} (\rightarrow I) \quad \frac{\Gamma \vdash_D M_1 : A_1 \rightarrow A_2 \quad \Gamma \vdash_D M_2 : A_1}{\Gamma \vdash_D M_1 M_2 : A_2} (\rightarrow E) \\ &\frac{\Gamma \vdash_D M : A}{\Gamma \vdash_D M : \forall X.A} (\forall I)^* \quad \frac{\Gamma \vdash_D M : \forall X.A}{\Gamma \vdash_D M[X := A_1] : A[X := A_1]} (\forall E) \end{aligned}$$

where  $(\forall I)^*$  denotes the eigenvariable condition  $X \notin \text{FV}(\Gamma)$ .

#### Definition 2 (Removing vacuous- $\forall$ )

1.  $\|x\| = x$ ,  $\|\lambda x : A.M\| = \lambda x : \|A\|. \|M\|$ ,  $\|M_1 M_2\| = \|M_1\| \|M_2\|$ ,
2.  $\|X\| = X$ ,  $\|A_1 \rightarrow A_2\| = \|A_1\| \rightarrow \|A_2\|$ ,  
 $\|\forall X.A\| = \forall X. \|A\|$  for  $X \in \text{FV}(A)$   
 $\|\forall X.A\| = \|A\|$  for  $X \notin \text{FV}(A)$
3.  $\|\Gamma\|(x) = \|\Gamma(x)\|$

If  $\|A\| = A$  then we say  $A$  has no vacuous  $\forall$ .

**Lemma 1** 1.  $\|A[X := B]\| = \|A\|[X := \|B\|]$

$$2. \|M[X := B]\| = \|M\|[X := \|B\|]$$

$$3. \text{FV}(A) = \text{FV}(\|A\|)$$

*Proof.* By induction on the structure of  $A$  or  $M$ . □

**Proposition 2** 1. If  $\Gamma \vdash_D M : A$  then  $\|\Gamma\| \vdash_D \|M\| : \|A\|$ .

2. If  $\Gamma \vdash_D M : A$  where each application of  $(\forall I)$  is not vacuous in the derivation, then for any  $\Gamma', M', A'$  with  $\|\Gamma'\| = \Gamma$ ,  $\|M'\| = M$ , and  $\|A'\| = A$  we have  $\Gamma' \vdash_D M' : A'$ .

*Proof.* First observe that given  $A$ , then any  $B$  such that  $\|B\| = A$  is generated by the following steps with fresh type variables  $\vec{Z}$ : (1) Case  $A = X$ :  $B = \forall \vec{Z}. X$ , (2) Case  $A = (A_1 \rightarrow A_2)$ :  $B = \forall \vec{Z}. (A'_1 \rightarrow A'_2)$ , (3) Case  $A = \forall X. A_1$ :  $B = \forall \vec{Z}. \forall X. A'_1$ . By induction on the derivation, we show only the case 2.

1. Case of  $\Gamma \vdash x : \Gamma(x)$  :

For any  $\Gamma', M', A'$  with  $\|\Gamma'\| = \Gamma$ ,  $\|M'\| = x$ , and  $\|A'\| = \Gamma(x)$ , we have  $M' \equiv x$  and  $A' \equiv \Gamma'(x)$ , and then we have  $\Gamma' \vdash x : \Gamma'(x)$ .

2.  $\Gamma \vdash MN : B$  from  $\Gamma \vdash M : A \rightarrow B$  and  $\Gamma \vdash N : A$  :

From the induction hypotheses, we have  $\Gamma' \vdash M' : C'$  for any  $\Gamma', M', C'$  such that  $\|\Gamma'\| = \Gamma$ ,  $\|M'\| = M$ ,  $\|C'\| = A \rightarrow B$ , and we have  $\Gamma' \vdash N' : A'$  for any  $\Gamma', N', A'$  such that  $\|\Gamma'\| = \Gamma$ ,  $\|N'\| = N$ ,  $\|A'\| = A$ . Here,  $C'$  should be in the form of  $\forall \vec{Z}.(A' \rightarrow B')$  where  $\|A'\| = A$  and  $\|B'\| = B$ . Then from  $\Gamma' \vdash M' : A' \rightarrow B'$  and  $\Gamma' \vdash N' : A'$ , we have  $\Gamma' \vdash M'N' : B'$  for any  $\Gamma', M'N', B'$  such that  $\|\Gamma'\| = \Gamma$ ,  $\|M'N'\| = MN$ ,  $\|B'\| = B$ .

3.  $\Gamma \vdash \lambda x:A.M : A \rightarrow B$  from  $\Gamma, x:A \vdash M : B$  :

From the induction hypothesis, we have  $\Gamma', x:A' \vdash M' : B'$  for any  $\Gamma', A', M', B'$  such that  $\|\Gamma', A'\| = \Gamma, A$ ,  $\|M'\| = M$ ,  $\|B'\| = B$ . Then we have  $\Gamma', x:A' \vdash \lambda x:A'.M' : \forall \vec{Z}.(A' \rightarrow B')$  where  $\|\Gamma'\| = \Gamma$ ,  $\|\lambda x:A'.M'\| = \lambda x:A.M$ , and  $\|\forall \vec{Z}.(A' \rightarrow B')\| = A \rightarrow B$ .

4.  $\Gamma \vdash M : \forall X.A$  from  $\Gamma \vdash M : A$  where  $X \notin \text{FV}(\Gamma)$  and  $X \in \text{FV}(A)$ :

From the induction hypothesis, we have  $\Gamma' \vdash M' : A'$  for any  $\Gamma', M', A'$  such that  $\|\Gamma'\| = \Gamma$ ,  $\|M'\| = M$ , and  $\|A'\| = A$ . Then from  $X \notin \text{FV}(\Gamma')$ , we have  $\Gamma' \vdash M' : \forall \vec{Z}.\forall X.A'$ , where  $\|\forall \vec{Z}.\forall X.A'\| = \forall X.\|A'\| = \forall X.A$ .

5.  $\Gamma \vdash M[X := B] : A[X := B]$  from  $\Gamma \vdash M : \forall X.A$  :

From the induction hypothesis, we have  $\Gamma' \vdash M' : \forall \vec{Z}.\forall X.A'$  for any  $\Gamma', M', A'$  such that  $\|\Gamma'\| = \Gamma$ ,  $\|M'\| = M$ , and  $\|A'\| = A$ . Take an arbitrary  $B'$  such that  $\|B'\| = B$ . Then we have  $\Gamma' \vdash M'[X := B'] : \forall \vec{Z}.A'[X := B']$  where  $\|M'[X := B']\| = \|M'\|[\|X := B'\|] = \|M'\|[\|X := B'\|] = M[X := B]$  and  $\|\forall \vec{Z}.A'[X := B']\| = \|A'\|[\|X := B'\|] = A[X := B]$ .  $\square$

**Lemma 2 (Permutation for bund variables)** If  $\Gamma \vdash_D M : \forall X.\forall Y.A$  then  $\Gamma \vdash_D M : \forall Y.\forall X.A$ .

**Lemma 3 (Substitution lemma 1)** If  $\Gamma \vdash_D M : A$  then  $\Gamma[X := B] \vdash_D M[X := B] : A[X := B]$ .

*Proof.* By induction on the derivation.  $\square$

**Lemma 4 (Substitution lemma 2)** If  $\Gamma, x:A \vdash_D M : B$  and  $\Gamma \vdash_D N : A$ , then  $\Gamma \vdash_D M[x := N] : B$ .

*Proof.* By induction on the first derivation.  $\square$

**Definition 3 (( $\forall I$ )( $\forall E$ )-reduction for (D))** Let  $X \notin \text{FV}(\Gamma)$ .

$$\frac{\frac{\Gamma \vdash M : A}{\Gamma \vdash M : \forall X.A} (\forall I)^*}{\Gamma \vdash M[X := B] : A[X := B]} (\forall E) \quad \mapsto \quad \Gamma \vdash M[X := B] : A[X := B]$$

Under this definition, we consider only derivations without ( $\forall I$ )( $\forall E$ )-redexes. This property is also called the INST-before-GEN property [11]. From now on, we consider derivations for  $\Gamma \vdash_D M : A$  with no vacuous  $\forall$  and the INST-before-GEN property. It is also remarked that ( $\forall E$ ) may be applied only after ( $\text{var}$ ), ( $\rightarrow E$ ), or ( $\forall E$ ).

**Definition 4 (Elimination-Introduction relation)** 1.  $A \leq_F^E B \stackrel{\text{def}}{\iff} \Gamma \vdash_D B$  is derived from  $\Gamma \vdash_D A$  by successive application of ( $\forall E$ ) including null application for some term.

2.  $A \leq_{I(\Gamma)} B \stackrel{\text{def}}{\iff} \Gamma \vdash_D B$  is derived from  $\Gamma \vdash_D A$  by successive application of ( $\forall I$ ) including null application for some term, where the eigenvariable condition holds w.r.t.  $\Gamma$ .

3.  $A \leq_{I(\Gamma)}^E B \stackrel{\text{def}}{\iff} A \leq_F^E C$  and  $C \leq_{I(\Gamma)} B$  for some type  $C$ .

For instance,  $\forall X.(X \rightarrow X) \leq_{I(\Gamma)}^E \forall X.\forall Y.((X \rightarrow Z \rightarrow Y) \rightarrow (X \rightarrow Z \rightarrow Y))$  where  $X, Y \notin \text{FV}(\Gamma)$ .

We also write  $\Gamma \vdash_D M : A \leq_{I(\Gamma)}^E N : B$ , if  $\Gamma \vdash_D M : A$  derives  $\Gamma \vdash_D N : B$  under the relation  $A \leq_{I(\Gamma)}^E B$ . In this case, we have  $M : A \leq_{I(\Gamma)}^E S(M) : B$  for some substitution  $S$  for type variables by the effect of application of  $(\forall E)$ .

**Lemma 5** ( $\leq_{I(\Gamma)}^E$ ) *Let  $m, n \geq 0$ , and neither  $A$  nor  $B$  has  $\forall$  as a top-symbol, and  $Y_1, \dots, Y_m \notin \text{FV}(\Gamma)$ .  $\forall X_1 \dots X_n.A \leq_{I(\Gamma)}^E \forall Y_1 \dots Y_m.B$  if and only if  $S(A) = B$  for some substitution  $S$  with  $\text{dom}(S) = \{X_1, \dots, X_n\}$ .*

*Proof.* ( $\Rightarrow$ ): Suppose  $\forall X_1 \dots X_n.A \leq_{I(\Gamma)}^E \forall Y_1 \dots Y_m.B$ . Then  $\forall X_1 \dots X_n.A \leq^E A[X_1 := A_1, \dots, X_n := A_n] = S(A) = B$  for some  $S$ , since  $B \leq_{I(\Gamma)} \forall Y_1 \dots Y_m.B$ . Hence,  $S(A) = B$  for some  $S$ .

( $\Leftarrow$ ): Suppose that  $S(A) = B$  for some  $S$ . Then  $\forall X_1 \dots X_n.A \leq^E S(A) = B \leq_{I(\Gamma)} \forall Y_1 \dots Y_m.B$  where each  $Y_i \notin \text{FV}(\Gamma)$ .  $\square$

**Remark 1** *Given  $A, B, \Gamma$ , then it is decidable to check whether  $A \leq_{I(\Gamma)}^E B$  holds or not.*

**Lemma 6 (partial order)** *Let  $A, B, C$  be types with no vacuous- $\forall$ .*

1.  $A \leq_{I(\Gamma)}^E A$
2. If  $A \leq_{I(\Gamma)}^E B$  and  $B \leq_{I(\Gamma)}^E C$  then  $A \leq_{I(\Gamma)}^E C$ .
3. If  $A \leq_{I(\Gamma)}^E B$  and  $B \leq_{I(\Gamma)}^E A$  then  $A \equiv B$ .

*Proof.* (2) If  $\Gamma \vdash_D A \leq_{I(\Gamma)}^E B$  and  $\Gamma \vdash_D B \leq_{I(\Gamma)}^E C$ , and then we have  $\Gamma \vdash_D A \leq_{I(\Gamma)}^E C$ . Moreover, if  $\Gamma \vdash_D M_1 : A \leq_{I(\Gamma)}^E M_2 : B$  and  $\Gamma \vdash_D M_2 : B \leq_{I(\Gamma)}^E M_3 : C$ , and then we have  $\Gamma \vdash_D M_1 : A \leq_{I(\Gamma)}^E M_3 : C$ .

(3) Let  $A = \forall X_1 \dots X_n.A'$  and  $B = \forall Y_1 \dots Y_m.B'$ , where  $X_1, \dots, X_n \in \text{FV}(A')$  and  $Y_1, \dots, Y_m \in \text{FV}(B')$ . Then  $S_1(A') = B'$  and  $S_2(B') = A'$  for some  $S_1, S_2$  with  $\text{dom}(S_1) = \{X_1, \dots, X_n\}$  and  $\text{dom}(S_2) = \{Y_1, \dots, Y_m\}$ . That is,  $A'$  and  $B'$  are variant, and hence  $S_1, S_2$  are bijective. Then  $n = m$  and  $\forall X_1 \dots X_n.A' \equiv \forall Y_1 \dots Y_m.B'$  under permutation for bound variables.  $\square$

Note that if we have vacuous- $\forall$ , then  $\forall XY.X \leq_I^E \forall Z.Z$  and  $\forall Z.Z \leq_I^E \forall XY.X$ , but  $\forall XY.X \neq \forall Z.Z$ .

**Lemma 7 (Generation lemma for System (D))** 1. *If  $\Gamma \vdash x : A$  then  $\Gamma(x) \leq_{I(\Gamma)}^E A$ .*

2. *If  $\Gamma \vdash \lambda x : A.M : B$ , then there exist  $B_1$  such that  $\Gamma, x : A \vdash M : B_1$  and  $A \rightarrow B_1 \leq_{I(\Gamma)} B$ .*
3. *If  $\Gamma \vdash M_1 M_2 : A$ , then there exist  $B_1, B_2, N_1$  such that  $\Gamma \vdash N_1 : B_1 \rightarrow B_2$  and  $\Gamma \vdash M_2 : B_1$  and  $N_1 M_2 : B_2 \leq_{I(\Gamma)}^E M_1 M_2 : A$ .*

*Proof.* By case analysis with the Elimination-Introduction property.

1. Suppose that  $\Gamma \vdash x : A$ .

We should start with  $\Gamma \vdash x : \Gamma(x)$ , and then the only way to derive  $\Gamma \vdash x : A$  is that  $\Gamma(x) \leq_{I(\Gamma)}^E A$ .

2. Suppose that  $\Gamma \vdash \lambda x : A_1.M : A_2$ .

Under the Elimination-Introduction property, the only way to derive  $\Gamma \vdash \lambda x : A_1.M : A_2$  is that  $\Gamma, x : A_1 \vdash M : B$  and  $A_1 \rightarrow B \leq_{I(\Gamma)} A_2$  for some  $B$ . Here, we cannot apply  $(\forall E)$  for  $A_1 \rightarrow B \leq_{I(\Gamma)}^E A_2$ .

3. Suppose that  $\Gamma \vdash M_1 M_2 : A$ .

Under the Elimination-Introduction property, the only way to derive  $\Gamma \vdash M_1 M_2 : A$  is that  $\Gamma \vdash N_1 : B_1 \rightarrow B_2$  and  $\Gamma \vdash M_2 : B_1$  and  $N_1 M_2 : B_2 \leq_{I(\Gamma)}^E M_1 M_2 : A$  for some  $N_1, B_1, B_2$ .

Here, we may apply  $(\forall E)$  for  $N_1 M_2 : B_2 \leq_{I(\Gamma)}^E M_1 M_2 : A$ , if  $B_2 = \forall \vec{X}.B'_2$  for some  $B'_2$ . Then  $\vec{X}$  cannot be free in  $N_2$  and hence  $N_2 \equiv M_2$ .  $\square$

**Definition 5** 1.  $(\lambda x:A.M)N \rightarrow_\beta M[x := N]$

2. If  $M \rightarrow_\beta N$  then  $RM \rightarrow_\beta RN$ ,  $MR \rightarrow_\beta NR$ , and  $\lambda x:A.M \rightarrow_\beta \lambda x:A.N$ .

**Lemma 8 (Abstraction)** If  $M \rightarrow_\beta N$  and  $S(M') = M$  for a substitution  $S$  for type variables, then there exists a term  $N'$  such that  $M' \rightarrow_\beta N'$  and  $S(N') = N$ .

*Proof.* By induction on the derivation of  $M \rightarrow_\beta N$ .  $\square$

**Proposition 3 (Subject reduction)** If  $\Gamma \vdash M : A$  and  $M \rightarrow_\beta N$ , then  $\Gamma \vdash N : A$ .

*Proof.* By induction on the derivation of  $M \rightarrow_\beta N$ , together with generation lemma.

- Case of  $\Gamma \vdash (\lambda x:A.M)N : B$  and  $(\lambda x:A.M)N \rightarrow M[x := N]$  :

$$\frac{\frac{\frac{\Gamma, x:A' \vdash M' : B_1}{\Gamma \vdash \lambda x:A'.M' : A' \rightarrow B_1} (\rightarrow I) \quad \Gamma \vdash N : A'}{\Gamma \vdash (\lambda x:A'.M')N : B_1 \leq_{I(\Gamma)}^E B} (\rightarrow E) \quad \Gamma \vdash M'[x := N] : B_1 \leq_{I(\Gamma)}^E B}{\Gamma \vdash (\lambda x:A.M)N : B} \mapsto \Gamma \vdash M[x := N] : B$$

where  $S(M') = M$  for some substitution  $S$ , and  $A' = A$  since if  $B = \forall \vec{X}.B'$  for some  $B'$  then  $\vec{X}$  cannot be free in  $A'$ .

- Case of  $\Gamma \vdash RM : B$  and  $RM \rightarrow RN$  from  $M \rightarrow N$ :

We also have  $R'M \rightarrow R'N$  from  $M \rightarrow N$  where  $S(R') = R$  for a substitution  $S$ .

$$\frac{\frac{\Gamma \vdash R' : B_2 \rightarrow B_1 \quad \Gamma \vdash M : B_2}{\Gamma \vdash R'M : B_1 \leq_{I(\Gamma)}^E B} (\rightarrow E)}{\Gamma \vdash RM : B}$$

From the induction hypothesis, we have  $\Gamma \vdash N : B_2$ , and then  $\Gamma \vdash R'N : B_1 \leq_{I(\Gamma)}^E RN : B$ .

- Case of  $\Gamma \vdash MR : B$  and  $MR \rightarrow NR$  from  $M \rightarrow N$  :

$$\frac{\frac{\Gamma \vdash M' : B_2 \rightarrow B_1 \quad \Gamma \vdash R : B_2}{\Gamma \vdash M'R : B_1 \leq_{I(\Gamma)}^E B} (\rightarrow E)}{\Gamma \vdash MR : B}$$

Since  $S(M') = M$  for some substitution  $S$ , we have  $M' \rightarrow N'$  and  $S(N') = N$  for some  $N'$ . From the induction hypothesis, we have  $\Gamma \vdash N' : B_2 \rightarrow B_1$ , and then  $\Gamma \vdash N'R : B_1 \leq_{I(\Gamma)}^E NR : B$ .

- Case of  $\Gamma \vdash \lambda x:A.M : B$  and  $\lambda x:A.M \rightarrow \lambda x:A.N$  from  $M \rightarrow N$  :

$$\frac{\frac{\Gamma, x:A \vdash M : B_1}{\Gamma \vdash \lambda x:A.M : A \rightarrow B_1 \leq_{I(\Gamma)} B} (\rightarrow I)}{\Gamma \vdash \lambda x:A.M : B}$$

From the induction hypothesis, we have  $\Gamma, x:A \vdash N : B_1$ , and then  $\Gamma \vdash \lambda x:A.N : A \rightarrow B_1 \leq_{I(\Gamma)} B$ .  $\square$

**Remark 2** If  $\lambda x:A.(Mx) : A \rightarrow B$ , then  $\lambda x:A.(Mx) \rightarrow_\eta M' : A_1 \rightarrow B_1$  that is a contravariant such that  $A \leq_I^E A_1$  and  $B_1 \leq_I^E B$ . For instance, we have  $x : (A \rightarrow \forall X.X) \vdash \lambda a:A.xa : A \rightarrow Z$ . Then we have  $\lambda a:A.xa \rightarrow_\eta x : A \rightarrow \forall X.X$ .



**Theorem 1 (Strong normalization)** *If  $\Gamma \vdash_D M : A$  then  $M$  is strongly normalizing.*

*Proof.* Suppose  $\Gamma \vdash_D M : A$  then  $\Gamma \vdash_{Cu} |M| : A$  and the Curry-term  $|M|$  is strongly normalizing, where  $|\cdot|$  is a forgetful mapping from (D)-terms to Curry-style terms. For (D)-terms  $M, N$ , if  $M \rightarrow_\beta N$  then  $|M| \rightarrow_\beta |N|$ .  $\square$

**Theorem 2 (Church-Rosser)**  *$\lambda 2$ -terms in the style of (D) are Church-Rosser with respect to  $\rightarrow_\beta$ .*

*Proof.* By the use of parallel reduction.  $\square$

**Remark 3** *Note that  $\lambda x:B.(\lambda x:A.x)x \rightarrow_\beta \lambda x:B.x$  and  $\lambda x:B.(\lambda x:A.x)x \rightarrow_\eta \lambda x:A.x$ . This implies that  $\rightarrow_\beta$  and  $\rightarrow_\eta$  are not commutative. Note also that well-typed terms are Church-Rosser w.r.t.  $\rightarrow_\beta$ , from the strong normalization property, weak Church-Rosser, and Newman's lemma. Another proof is that type-annotated terms in the style of (D) are Church-Rosser together with the subject reduction property.*

**Proposition 4 (Reductions between type-related problems)** 1.  $TCP \hookrightarrow TIP$ :

$\Gamma \vdash M : A$  iff  $\Gamma, z : (A \rightarrow Z) \vdash zM : B$  for some  $B$ , where  $z, Z$  are fresh variables.

2.  $TIP \hookrightarrow TCP$ :

$\Gamma \vdash M : B$  for some  $B$  iff  $\Gamma, z : \forall X.(X \rightarrow Z) \vdash zM : Z$ , where  $z, Z$  are fresh variables.

3.  $TIP \hookrightarrow TP$ : Let  $\Gamma = \{x_1 : A_1, \dots, x_n : A_n\}$  and  $\text{Dom}(\Gamma) = \text{FV}(M)$ .

$\Gamma \vdash M : B$  for some  $B$  iff  $\Sigma \vdash \lambda x_1:A_1 \dots \lambda x_n:A_n.M : B$  for some  $\Sigma, B$ .

**Definition 6 (Normal forms of (D)-terms)**

$$\begin{aligned} N \in \text{NF} &::= V \mid \lambda x:A.N \\ V &::= x \mid V N \end{aligned}$$

**Proposition 5** *Let  $N \in \text{NF}$ . If  $\Gamma \vdash_D M : A$  with the the Elimination-Introduction property, then each application of the rule  $(\forall E)$  in the derivation can be restricted to the following form:*

$$\frac{\Gamma \vdash N : \forall X.B}{\Gamma \vdash N : A[X := B]} (\forall E')$$

*Proof.* By induction on the derivation of normal forms, together with the generation lemma.

1. Case of  $\Gamma \vdash \lambda x:A.N : B$

From the generation lemma, we have the following derivation:

$$\frac{\frac{\Gamma, x:A \vdash N : C}{\Gamma \vdash \lambda x:A.N : A \rightarrow C \leq_{I(\Gamma)} B}}{\Gamma \vdash \lambda x:A.N : B}$$

From the induction hypothesis, we have a derivation for  $\Gamma, x:A \vdash N : B$ , where the derivation may contain only  $(\forall E')$  instead of  $(\forall E)$ .

2. Case of  $\Gamma \vdash xN_1 \dots N_n : B$

From the generation lemma, for some  $A_1, B_2$  we have  $x : \Gamma(x) \leq^E A_1 \rightarrow B_2$  where  $(\forall E')$  may be applied, and  $\Gamma \vdash N_1 : A_1$  where each application of  $(\forall E)$  can be restricted to  $(\forall E')$  by the induction hypothesis. Then for some  $A_2, B_3, N'_1$ , we have  $\Gamma \vdash xN_1 : B_2 \leq^E xN'_1 : A_2 \rightarrow B_3$  and  $\Gamma \vdash N_2 : A_2$ . Here,  $xN'_1 : A_2 \rightarrow B_3$  is obtained from  $xN_1 : B_2$  by consecutive application of  $(\forall E)$ . That is,  $B_2$  is in the form of  $\forall \vec{X}_2.B'_2$  for some  $B'_2$ , and  $\vec{X}_2$  cannot appear in  $N_1$  as free

type variables. Hence, a chain of applications of  $(\forall E)$  can be replaced with  $(\forall E')$ , so that we have  $xN'_1 = xN_1$ . In addition,  $(\forall E)$  can be restricted to  $(\forall E')$  in the derivation of  $\Gamma \vdash N_2 : A_2$  by the induction hypothesis. Following this argument, we have a chain of applications of  $(\forall E)$ :

$\Gamma(x) \leq^E (A_1 \rightarrow B_2), B_2 = \forall \vec{X}_2. B'_2 \leq^E (A_2 \rightarrow B_3), \dots, B_n = \forall \vec{X}_n. B'_n \leq^E (A_n \rightarrow B_{n+1})$ , such that

$x : \Gamma(x) \leq^{E'} x : (A_1 \rightarrow \forall \vec{X}_2. B'_2)$  and  $N_1 : A_1$  where  $\vec{X}_2 \notin \text{FV}(N_1)$ ,

$xN_1 : \forall \vec{X}_2. B'_2 \leq^{E'} xN_1 : (A_2 \rightarrow \forall \vec{X}_3. B'_3)$  and  $N_2 : A_2$  where  $\vec{X}_3 \notin \text{FV}(N_1 N_2)$ ,

$\dots$ ,

$xN_1 \dots N_{n-1} : \forall \vec{X}_n. B'_n \leq^{E'} (A_n \rightarrow \forall \vec{X}_{n+1}. B'_{n+1})$  and  $N_n : A_n$  where  $\vec{X}_{n+1} \notin \text{FV}(N_1 \dots N_n)$ , and

$xN_1 \dots N_n : \forall \vec{X}_{n+1}. B'_{n+1} \leq^{E'}_{I(\Gamma)} xN_1 \dots N_n : B$ .

Thus, each application of  $(\forall E)$  in the derivation of  $\Gamma \vdash xN_1 \dots N_n : B$  can be replaced with  $(\forall E')$ .  $\square$

We divide the set of type variables into two countable sets: TVars for the usual type variables and UVars for type variables called unification variable.

$$\text{TypeVars} = \text{TVars} \cup \text{UVars}$$

The syntax of output types  $\hat{A}$  of type inference is defined as follows:

$$\hat{A}, \hat{B} \in \text{Output} ::= X \mid \alpha \mid (\hat{A} \rightarrow \hat{B}) \mid \forall X. \hat{A}$$

where  $X \in \text{TVars}$  is a type variable,  $\alpha \in \text{UVars}$  is a type variable also called a unification variable.

A unification procedure for the multiset  $E$  of unification equations is defined as usual by the following transformation rules, which give a most general unifier:

1.  $\{\hat{A} \doteq \hat{A}\} \cup E \Longrightarrow E$
2.  $\{\alpha \doteq \hat{A}\} \cup E \Longrightarrow \{\alpha \doteq \hat{A}\} \cup E[\alpha := \hat{A}]$  if  $\alpha \notin \text{UVars}(\hat{A})$
3.  $\{\hat{A}_1 \rightarrow \hat{A}_2 \doteq \hat{B}_1 \rightarrow \hat{B}_2\} \cup E \Longrightarrow \{\hat{A}_1 \doteq \hat{B}_1, \hat{A}_2 \doteq \hat{B}_2\} \cup E$
4.  $\{\forall X. \hat{A} \doteq \forall X. \hat{B}\} \cup E \Longrightarrow \{\hat{A} \doteq \hat{B}\} \cup E$

Here, we consider type inference of terms in the style of (D) where a given term is a normal form.

**Definition 7 (Type inference for (D): non-deterministic version for normal case)**

1.  $\text{type}(\Gamma; x) = \Gamma(x)$
2.  $\text{type}(\Gamma; \lambda x. A.M) = (A \rightarrow B)$ , where  $\text{type}(\Gamma, x : A; M) \leq^E_{I(\Gamma, A)} B$
3.  $\text{type}(\Gamma; M_1 M_2) = B_2$ , where  $\text{type}(\Gamma; M_1) \leq^E B_1 \rightarrow B_2$  and  $\text{type}(\Gamma; M_2) \leq^E_{I(\Gamma)} B_1$  for some  $B_1$

As a shorthand, we write  $\vec{x} : \vec{A}$  for  $x_1 : A_1 \dots x_n : A_n$ , and  $\forall \vec{X}. A$  for  $\forall X_1 \dots X_n. A$  ( $n \geq 0$ ). By deleting  $\forall \vec{X}$  at strictly positive positions, we use the following notation  $\succeq$ :

$$\begin{aligned} \forall \vec{X}_1 (A_1 \rightarrow \forall \vec{X}_2 (A_2 \rightarrow \dots \rightarrow \forall \vec{X}_n (A_n \rightarrow A) \dots)) &\succeq (A_1 \rightarrow \forall \vec{X}_2 (A_2 \rightarrow \dots \rightarrow \forall \vec{X}_n (A_n \rightarrow A) \dots)) \\ &\succeq (A_1 \rightarrow (A_2 \rightarrow \dots \rightarrow \forall \vec{X}_n (A_n \rightarrow A) \dots)) \succeq \dots \succeq (A_1 \rightarrow (A_2 \rightarrow \dots \rightarrow (A_n \rightarrow A) \dots)). \end{aligned}$$

**Definition 8 (Type-inference for (D): deterministic version for normal case)**

1.  $\text{type}(\Gamma; x) = \Gamma(x)$
2.  $\text{type}(\Gamma; \lambda \vec{x}. \vec{A}. V) = (\vec{A} \rightarrow \text{type}(\Gamma, \vec{x} : \vec{A}; V))$  where  $\vec{x} : \vec{A}$  denotes  $x_1 : A_1 \dots x_n : A_n$  ( $n \geq 1$ )

3.  $\text{type}(\Gamma; xN_1 \dots N_n) = A[\vec{X} := \vec{B}]$ , where we set

$\Gamma(x) = \forall \vec{X}_1 (A_1 \rightarrow \forall \vec{X}_2 (A_2 \rightarrow \dots \rightarrow \forall \vec{X}_n (A_n \rightarrow A) \dots))$ ,  $\vec{X} = \vec{X}_1 \dots \vec{X}_n$ , and  $\vec{B} = \vec{B}_1 \dots \vec{B}_n$  ( $n \geq 1$ )

(a) Case of  $N_1 = V_1$ :

There exist some  $\vec{B}_1$  such that  $\text{type}(\Gamma; V_1) \leq_{I(\Gamma)}^E A_1[\vec{X}_1 := \vec{B}_1]$ .

(b) Case of  $N_1 = \lambda \vec{y} : \vec{C}. V_1$  where  $\vec{C} = C_1, \dots, C_k$  ( $k \geq 1$ ):

Let  $(\vec{C} \rightarrow \text{type}(\Gamma, \vec{y} : \vec{C}; V_1))$  be  $\text{type}(\Gamma; N_1)$ . There exist some  $\vec{B}_1, D_1$  such that  $A_1[\vec{X}_1 := \vec{B}_1] \succeq (C_1 \rightarrow \dots \rightarrow C_k \rightarrow D_1)$  and  $\text{type}(\Gamma, \vec{y} : \vec{C}; V_1) \leq_{I(\Gamma, \vec{C})}^E D_1$ .

(c) Case of  $N_i = V_i$  ( $1 < i \leq n$ ):

There exist some  $\vec{B}_i$  such that  $\text{type}(\Gamma; V_i) \leq_{I(\Gamma)}^E A_i[\vec{X}_1 \dots \vec{X}_i := \vec{B}_1 \dots \vec{B}_i]$ .

(d) Case of  $N_i = \lambda \vec{y} : \vec{C}. V_i$  ( $1 < i \leq n$ ) where  $\vec{C} = C_1, \dots, C_k$  ( $k \geq 1$ ):

Let  $(\vec{C} \rightarrow \text{type}(\Gamma, \vec{y} : \vec{C}; V_i))$  be  $\text{type}(\Gamma; N_i)$ . There exist some  $\vec{B}_i, D_i$  such that  $A_i[\vec{X}_1 \dots \vec{X}_i := \vec{B}_1 \dots \vec{B}_i] \succeq (C_1 \rightarrow \dots \rightarrow C_k \rightarrow D_i)$  and  $\text{type}(\Gamma, \vec{x} : \vec{C}; V_i) \leq_{I(\Gamma, \vec{C})}^E D_i$ .

**Remark 4** 1. Although the cases of  $N_1$  are included in those of  $N_i$  ( $i \geq 1$ ), we write the first cases for readability.

2. We use the notation  $\vec{A} \rightarrow \text{type}(\Gamma, \vec{x} : \vec{A}; V)$  for  $\text{type}(\Gamma; \lambda \vec{x} : \vec{A}. V)$ . If a given term is in the form of  $\lambda \vec{x} : \vec{A}. V$ , then the expression  $\vec{A} \rightarrow \text{type}(\Gamma, \vec{x} : \vec{A}; V)$  simply means that  $A_1 \rightarrow \dots \rightarrow A_n \rightarrow \text{type}(\Gamma, \vec{x} : \vec{A}; V)$  where  $\vec{A} = (A_1, \dots, A_n)$ .

**Lemma 9** 1. It is decidable to verify whether the condition in the case of  $N = V$  of type, i.e.,  $\text{type}(\Gamma; N) \leq_{I(\Gamma)}^E A[\vec{X} := \vec{B}]$  for some  $\vec{B}$ , holds or not.

2. It is decidable to verify whether the condition in the case of  $N = \lambda \vec{y} : \vec{C}. V$  of type, i.e.,  $A[\vec{X} := \vec{B}] \succeq (C_1 \rightarrow \dots \rightarrow C_k \rightarrow D)$  for some  $\vec{B}, D$  such that  $\text{type}(\Gamma, \vec{y} : \vec{C}; V) \leq_{I(\Gamma, \vec{C})}^E D$ , holds or not.

*Proof.* 1. The condition that  $\text{type}(\Gamma; N) \leq_{I(\Gamma)}^E A[\vec{X} := \vec{B}]$  for some  $\vec{B}$  can be verified by first order unification as follows, see also Lemma 5: Let  $\forall \vec{Y}. C = \text{type}(\Gamma; N)$  ( $C$  has no  $\forall$  as a top-symbol),  $\forall \vec{Z}. A' = A$  ( $A'$  has no  $\forall$  as a top-symbol), and  $\vec{\alpha}, \vec{\beta}$  be fresh unification variables. Then solve the unification equation such that  $C[\vec{Y} := \vec{\beta}] \doteq A'[\vec{X} := \vec{\alpha}]$ . If the unification equation is solvable under a unifier  $S$ , then we set  $\vec{B} = S(\vec{\alpha})$ .

2. The condition that  $A[\vec{X} := \vec{B}] \succeq (C_1 \rightarrow \dots \rightarrow C_k \rightarrow D)$  for some  $\vec{B}, D$  can be verified by first order unification, as follows: Let  $\vec{\beta}, \delta$  be fresh unification variables, and  $A'$  be obtained from  $A$  by removing  $\forall \vec{X}$  at strictly positive positions just like that  $\forall \vec{X}_1 (A_1 \rightarrow \forall \vec{X}_2 (A_2 \rightarrow \dots \rightarrow \forall \vec{X}_n (A_n \rightarrow A) \dots)) \succeq (A_1 \rightarrow (A_2 \rightarrow \dots \rightarrow (A_n \rightarrow A) \dots))$ . Then solve the unification equation such that  $A'[\vec{X} := \vec{\beta}] \doteq (C_1 \rightarrow \dots \rightarrow C_k \rightarrow \delta)$ . If the unification equation is solvable under a unifier  $S$ , then we can check whether  $\text{type}(\Gamma, \vec{x} : \vec{C}; V) \leq_{I(\Gamma, \vec{C})}^E S(\delta)$  as in the previous case. Let  $\forall \vec{Y}. E$  be  $\text{type}(\Gamma, \vec{x} : \vec{C}; V)$ , and  $\vec{\gamma}$  be fresh unification variables. Then solve the unification equation  $E[\vec{Y} := \vec{\gamma}] \doteq S(\delta)$ . Now suppose that the equation is solvable under a unifier  $T$ . Next, we recover  $\forall \vec{X}$  to be removed for  $\succeq$  under the variable conditions  $I(\Gamma), I(\Gamma, \vec{C}_1), \dots, I(\Gamma, \vec{C}_1, \dots, \vec{C}_{k-1})$ . Finally, we set  $\vec{B} = T(S(\vec{\beta}))$  and  $D = T(S(\delta))$ .  $\square$

**Proposition 6 (Soundness of type)** If  $\text{type}(\Gamma; N) = A$  then we have  $\Gamma \vdash N : A$ .

*Proof.* The soundness is proved by induction on the length of a term.

1. Case  $N$  of  $x$ :

We always have  $\Gamma \vdash x : \text{type}(\Gamma; x)$ .

2. Case  $N$  of  $\lambda \vec{x} : \vec{A}. V$ :

From the induction hypothesis, we have  $\Gamma, \vec{x} : \vec{A} \vdash V : \text{type}(\Gamma, \vec{x} : \vec{A}; V)$ . Then  $\Gamma \vdash \lambda \vec{x} : \vec{A}. V : (\vec{A} \rightarrow \text{type}(\Gamma, \vec{x} : \vec{A}; V))$ , and  $\text{type}(\Gamma; \lambda \vec{x} : \vec{A}. V) = (\vec{A} \rightarrow \text{type}(\Gamma, \vec{x} : \vec{A}; V))$ .

3. Case  $N$  of  $xN_1 \dots N_n$ :

Let  $\Gamma(x) = \forall \vec{X}_1 (A_1 \rightarrow \forall \vec{X}_2 (A_2 \rightarrow \dots \rightarrow \forall \vec{X}_n (A_n \rightarrow A) \dots))$ , and  $\vec{X} = \vec{X}_1 \dots \vec{X}_n$ .

(a) Case  $N_i$  of  $V_i$  ( $1 \leq i \leq n$ ):

From the induction hypothesis, we have  $\Gamma \vdash V_i : \text{type}(\Gamma; V_i)$ , and from the assumption,  $\text{type}(\Gamma; V_i) \leq_{I(\Gamma)}^E A_i[\vec{X}_1 \dots \vec{X}_i := \vec{B}_1 \dots \vec{B}_i]$  and we also have  $\Gamma \vdash x : \Gamma(x) \leq^E (A_i \rightarrow \forall \vec{X}_{i+1} (A_{i+1} \rightarrow \dots \rightarrow \forall \vec{X}_n (A_n \rightarrow A) \dots))[\vec{X}_1 \dots \vec{X}_i := \vec{B}_1 \dots \vec{B}_i]$  for some  $\vec{B}_i$ . Then we have  $\Gamma \vdash xN_1 \dots N_i : \forall \vec{X}_{i+1} (A_{i+1} \rightarrow \dots \rightarrow \forall \vec{X}_n (A_n \rightarrow A) \dots)[\vec{X}_1 \dots \vec{X}_i := \vec{B}_1 \dots \vec{B}_i]$ .

(b) Case  $N_i$  of  $\lambda \vec{y} : \vec{C}. V_i$  ( $1 \leq i \leq n$ ):

From the induction hypothesis, we have  $\Gamma, \vec{y} : \vec{C} \vdash V_i : \text{type}(\Gamma, \vec{y} : \vec{C}; V_i)$ , and from the assumption, we also have  $A_i[\vec{X}_1 \dots \vec{X}_i := \vec{B}_1 \dots \vec{B}_i] \succeq (C_1 \rightarrow \dots \rightarrow C_k \rightarrow D_i)$  and  $\text{type}(\Gamma, \vec{y} : \vec{C}; V_i) \leq_{I(\Gamma, \vec{C})}^E D_i$  for some  $\vec{B}_i, D_i$ . Then from the induction hypothesis, we have  $\Gamma, \vec{y} : \vec{C} \vdash V_i : \text{type}(\Gamma, \vec{y} : \vec{C}; V_i)$ , and moreover  $\Gamma \vdash N_i : \forall \vec{Z}_1 (C_1 \rightarrow \dots \rightarrow \forall \vec{Z}_k (C_k \rightarrow D_i))$  under the variable condition, where each  $\forall \vec{Z}_i$  is the deleted quantifiers on the condition that  $A_i[\vec{X}_1 \dots \vec{X}_i := \vec{B}_1 \dots \vec{B}_i] \succeq (C_1 \rightarrow \dots \rightarrow C_k \rightarrow D_i)$ . Here, we have  $\forall \vec{Z}_1 (C_1 \rightarrow \dots \rightarrow \forall \vec{Z}_k (C_k \rightarrow D_i)) = A_i[\vec{X}_1 \dots \vec{X}_i := \vec{B}_1 \dots \vec{B}_i]$ . Hence, we have  $\Gamma \vdash xN_1 \dots N_i : \forall \vec{X}_i (A_i \rightarrow \dots \rightarrow \forall \vec{X}_n (A_n \rightarrow A) \dots)[\vec{X}_1 \dots \vec{X}_i := \vec{B}_1 \dots \vec{B}_i]$ .

In this way, we have  $\Gamma \vdash xN_1 \dots N_n : A[\vec{X}_1 \dots \vec{X}_n := \vec{B}_1 \dots \vec{B}_n]$  and  $\text{type}(\Gamma; xN_1 \dots N_n) = A[\vec{X}_1 \dots \vec{X}_n := \vec{B}_1 \dots \vec{B}_n]$ .  $\square$

**Proposition 7 (Completeness of type)** *Given a context  $\Gamma$  and a normal term  $N$ , let  $A$  be a type such that  $\Gamma \vdash N : A$ . Then we have  $\text{type}(\Gamma; V) \leq_{I(\Gamma)}^E A$  if  $N = V$ , and  $A \succeq (B_1 \rightarrow \dots \rightarrow B_n \rightarrow C)$  for some  $C$  such that  $\text{type}(\Gamma, \vec{x} : \vec{B}; V) \leq_{I(\Gamma, \vec{B})}^E C$  if  $N = \lambda \vec{x} : \vec{B}. V$ .*

*Proof.* The completeness is proved by induction on the derivation with the generation lemma and the Elimination-Introduction property.

1. We have  $\Gamma \vdash x : \Gamma(x) \leq_{I(\Gamma)}^E \Gamma(x)$ .2.  $\Gamma \vdash xN_1 \dots N_n : A$ 

Let  $\Gamma(x) = \forall \vec{X}_1 (A_1 \rightarrow \forall \vec{X}_2 (A_2 \rightarrow \dots \rightarrow \forall \vec{X}_n (A_n \rightarrow A_0) \dots))$ , and  $\vec{X} = \vec{X}_1 \dots \vec{X}_n$ . Then from the generation lemma, we have  $\Gamma \vdash xN_1 \dots N_n : A_0[\vec{X} := \vec{B}] \leq_{I(\Gamma)}^E A$  for some  $\vec{B}$ , where  $\text{type}(\Gamma; xN_1 \dots N_n) = A_0[\vec{X} := \vec{B}]$ .

3.  $\Gamma \vdash \lambda \vec{x} : \vec{C}. V : A$ 

From the generation lemma, we have  $\Gamma \vdash \lambda \vec{x} : \vec{C}. V : C_1 \rightarrow A_1 \leq_{I(\Gamma)}^E A$  for some  $A_1$ , such that  $\Gamma, x_1 : C_1 \vdash \lambda \vec{x}' : \vec{C}'. V : A_1$ . Then we also have  $\Gamma, x_1 : C_1 \vdash \lambda \vec{x}' : \vec{C}'. V : C_2 \rightarrow A_2 \leq_{I(\Gamma, C_1)}^E A_1$  for some  $A_2$ . Following similar reasoning, we have  $\Gamma, x_1 : C_1, \dots, x_n : C_n \vdash \lambda x_n : C_n. V : C_n \rightarrow$

$A_n \leq_{I(\Gamma, C_1, \dots, C_{n-1})} A_{n-1}$  for some  $A_n$ , such that  $\Gamma, \vec{x} : \vec{C} \vdash V : A_n$  where  $\text{type}(\Gamma, \vec{C}; V) \leq_{I(\Gamma, \vec{C})}^E A_n$  by the induction hypothesis. Now we have the following relations:

$$\begin{array}{ccc} C_n \rightarrow A_n & \leq_{I(\Gamma, C_1, \dots, C_{n-1})} & A_{n-1} \\ \dots & \vdots & \dots \\ C_2 \rightarrow A_2 & \leq_{I(\Gamma, C_1)} & A_1 \\ C_1 \rightarrow A_1 & \leq_{I(\Gamma)} & A \end{array}$$

Namely there are some quantifiers  $\forall \vec{X}_i$ , such that  $A = \forall \vec{X}_1.(C_1 \rightarrow A_1)$ ,  $A_1 = \forall \vec{X}_2.(C_2 \rightarrow A_2)$ ,  $\dots$ , and  $A_{n-1} = \forall \vec{X}_n.(C_n \rightarrow A_n)$ . Hence, we have  $A \succeq (C_1 \rightarrow \dots \rightarrow C_n \rightarrow A_n)$  and  $\text{type}(\Gamma, \vec{C}; V) \leq_{I(\Gamma, \vec{C})}^E A_n$ .

4.  $\Gamma \vdash N : \forall X.A$  from  $\Gamma \vdash N : A$  where  $X \notin \text{FV}(\Gamma)$

(a) Case  $N$  of  $V$ :

From the induction hypothesis, we have  $\text{type}(\Gamma; V) \leq_{I(\Gamma)}^E A$  and  $A \leq_{I(\Gamma)}^E \forall X.A$ . Then  $\text{type}(\Gamma; V) \leq_{I(\Gamma)}^E \forall X.A$ .

(b) Case  $N$  of  $\lambda \vec{x} : \vec{B}. V$ :

From the induction hypothesis, we have  $A \succeq (B_1 \rightarrow \dots \rightarrow B_n \rightarrow C)$  for some  $C$  such that  $\text{type}(\Gamma, \vec{x} : \vec{B}; V) \leq_{I(\Gamma, \vec{B})}^E C$ . Then we also have  $\forall X.A \succeq (B_1 \rightarrow \dots \rightarrow B_n \rightarrow C)$ .

5.  $\Gamma \vdash N : A[X := D]$  from  $\Gamma \vdash N : \forall X.A$

(a) Case  $N$  of  $V$ :

From the induction hypothesis, we have  $\text{type}(\Gamma; V) \leq_{I(\Gamma)}^E \forall X.A$  and  $\forall X.A \leq_{I(\Gamma)}^E A[X := D]$ . Then  $\text{type}(\Gamma; V) \leq_{I(\Gamma)}^E A[X := D]$  from the transitivity.

(b) Case  $N$  of  $\lambda \vec{x} : \vec{B}. V$ :

This case is impossible under the Elimination-Introduction property, since  $\Gamma \vdash \lambda \vec{x} : \vec{B}. V : \forall X.A$  should be introduced by  $(\forall I)$ .  $\square$

Next, we define a type inference algorithm in general. For this, the notion of generalization of types is introduced.

**Definition 9 (Generalization)** Given a type  $A$ , then define the set of generalization of  $A$ , denoted by  $\text{Gen}(A)$  such that for each  $P \in \text{Gen}(A)$ , we have  $S(P) \equiv A$  for some substitution  $S$ .

1.  $\text{Gen}_\Delta(X) = \{X^{id}\}$  if  $X \notin \Delta$
2.  $\text{Gen}_\Delta(X) = \{X\}$  if  $X \in \Delta$
3.  $\text{Gen}_\Delta(A \rightarrow B) = \{Z^{[Z:=A \rightarrow B]}\} \cup \{P_1 \rightarrow P_2 \mid P_1 \in \text{Gen}_\Delta(A_1), P_2 \in \text{Gen}_\Delta(A_2)\} \cup \text{merge}(\text{Gen}_\Delta(A_1), \text{Gen}_\Delta(A_2))$   
where  $Z$  is a fresh variable, if  $\text{FV}(A \rightarrow B) \not\subseteq \Delta$
4.  $\text{Gen}_\Delta(A \rightarrow B) = \{P_1 \rightarrow P_2 \mid P_1 \in \text{Gen}_\Delta(A_1), P_2 \in \text{Gen}_\Delta(A_2)\} \cup \text{merge}(\text{Gen}_\Delta(A_1), \text{Gen}_\Delta(A_2))$   
where  $Z$  is a fresh variable, if  $\text{FV}(A \rightarrow B) \subseteq \Delta$
5.  $\text{Gen}_\Delta(\forall X.A) = \{Z^{[Z:=\forall X.A]}\} \cup \{\forall X.P \mid P \in \text{Gen}_{\Delta \cup \{X\}}(A)\}$  where  $Z$  is fresh, if  $\text{FV}(\forall X.A) \not\subseteq \Delta$
6.  $\text{Gen}_\Delta(\forall X.A) = \{\forall X.P \mid P \in \text{Gen}_{\Delta \cup \{X\}}(A)\}$ , if  $\text{FV}(\forall X.A) \subseteq \Delta$

7.  $\text{merge}(\text{Gen}_\Delta(A), \text{Gen}_\Delta(B)) = \{P_A \rightarrow P_B \mid P_1 \text{ contains } Z_1^{[Z_1:=C]} \text{ and } P_2 \text{ contains } Z_2^{[Z_2:=C]}$   
 for some  $P_1 \in \text{Gen}_\Delta(A)$  and  $P_2 \in \text{Gen}_\Delta(B)$ , and  
 $P_A$  is obtained from  $P_1$  by replacing some occurrences of  $Z_1^{[Z_1:=C]}$  in  $P_1$  with  $Z^{[Z:=C]}$ , and  
 $P_B$  is obtained from  $P_2$  by replacing some occurrences of  $Z_2^{[Z_2:=C]}$  in  $P_2$  with  $Z^{[Z:=C]}$ ,  
 where  $Z$  is a fresh variable}

Here,  $\Delta$  in  $\text{Gen}_\Delta(A)$  denotes the set of bound type-variables in  $\text{FV}(A)$ , such that for each  $X \in \Delta$  we have some context  $C \neq []$  with  $\forall X. C[A]$ .

Given a term  $M$ , and we write  $\text{Atype}(M)$  for the multiset of annotated types in  $M$ , to say  $[A_1, \dots, A_n]$ . Then we have generalizations of each type  $[\text{Gen}(A_1), \dots, \text{Gen}(A_n)]$ .

Next define the set of terms, denoted by  $\text{Gen}(M)$ , such that  $\text{Gen}(M) = \{M[Z_1, \dots, Z_n] \mid Z_1 \in \text{Gen}(A_1), \dots, Z_n \in \text{Gen}(A_n)\}$ , where  $M[Z_1, \dots, Z_n]$  is a term obtained from  $M$  by replacing each occurrence  $A_i$  in  $M$  with  $Z_i \in \text{Gen}(A_i)$ . For each term  $N \in \text{Gen}(M)$  we have  $S(N) = M$  for some substitution  $S$  for type variables in  $N$ . That is, each term  $N \in \text{Gen}(M)$  is a term where annotated types in  $M$  are generalized.

We show some examples, where we may omit the identity substitution  $\text{id}$ .

- $\text{Gen}(X \rightarrow Y) = [(X^{\text{id}} \rightarrow Y^{\text{id}}), Z^{[Z:=X \rightarrow Y]}]$
- $\text{Gen}((X \rightarrow X) \rightarrow X \rightarrow X) =$   
 $[(X \rightarrow X) \rightarrow X \rightarrow X, Z_1^{[Z_1:=X \rightarrow X]} \rightarrow X \rightarrow X, (X \rightarrow X) \rightarrow Z_2^{[Z_2:=X \rightarrow X]},$   
 $Z^{[Z:=X \rightarrow X]} \rightarrow Z^{[Z:=X \rightarrow X]}, Z_1^{[Z_1:=X \rightarrow X]} \rightarrow Z_2^{[Z_2:=X \rightarrow X]}, Z_3^{[Z_3:=(X \rightarrow X) \rightarrow X \rightarrow X]}]$
- $\text{Gen}(\forall X.(X \rightarrow X)) = [\forall X.(X \rightarrow X), Z^{[Z:=\forall X.(X \rightarrow X)}]$   
 $\text{Gen}(\forall X.(X \rightarrow Y \rightarrow Y)) = [\forall X.(X \rightarrow Y \rightarrow Y), \forall X.(X \rightarrow Z^{[Z:=Y \rightarrow Y]}), Z^{[Z:=\forall X.(X \rightarrow Y \rightarrow Y)}]$
- Let  $B \equiv (\forall X.(X \rightarrow X)) \rightarrow \forall X.(X \rightarrow X)$ .  
 $\text{Gen}(B) = [(\forall X.(X \rightarrow X)) \rightarrow \forall X.(X \rightarrow X), Z_1^{[Z_1:=\forall X.X \rightarrow X]} \rightarrow \forall X.(X \rightarrow X),$   
 $Z_1^{[Z_1:=\forall X.X \rightarrow X]} \rightarrow Z_2^{[Z_2:=\forall X.X \rightarrow X]}, Z^{[Z:=\forall X.X \rightarrow X]} \rightarrow Z^{[Z:=\forall X.X \rightarrow X]},$   
 $\forall X.(X \rightarrow X) \rightarrow Z_2^{[Z_2:=\forall X.X \rightarrow X]}, Z_3^{[Z_3:=(\forall X.(X \rightarrow X)) \rightarrow \forall X.(X \rightarrow X)}]$

Note that  $\text{Gen}_\Delta(A)$  is a finite set of types, and then  $\text{Gen}_\Delta(M)$  is also a finite set of terms. We always have  $A \in \text{Gen}_\Delta(A)$  and  $\text{id}(A) = A$ , and hence  $M \in \text{Gen}_\Delta(M)$ .

**Definition 10 (Type inference for Curry with explicit domains: Non-deterministic version)**

1.  $\text{Type}(\Gamma; x) = \Gamma(x)$
2.  $\text{Type}(\Gamma; \lambda x:A.M) = (A \rightarrow B)$ , where  $\text{Type}(\Gamma, x:A; N) \leq_{I(\Gamma, A)}^E M : B$  for some  $N \in \text{Gen}(M)$
3.  $\text{Type}(\Gamma; M_1 M_2) = B_2$ , where  $\text{Type}(\Gamma; N_1) \leq^E M_1 : B_1 \rightarrow B_2$  and  $\text{Type}(\Gamma; N_2) \leq_{I(\Gamma)}^E M_2 : B_1$  for some  $B_1$  and some  $N_1 \in \text{Gen}(M_1), N_2 \in \text{Gen}(M_2)$

**Proposition 8 (Soundness and completeness of non-deterministic Type)**

1. If  $\text{Type}(\Gamma; M) = A$  then  $\Gamma \vdash M : A$ .
2. Given a context  $\Gamma$  and a term  $M$ , let  $A$  be a type such that  $\Gamma \vdash M : A$ . Then we have  $\text{Type}(\Gamma; N) \leq_{I(\Gamma)}^E M : A$  for some  $N \in \text{Gen}(M)$ .

*Proof.* The soundness is proved by induction on the length of  $M$ .

1.  $\text{Type}(\Gamma; x) = \Gamma(x)$ :

We have  $\Gamma \vdash x : \Gamma(x)$ .

2.  $\text{Type}(\Gamma; \lambda x : A.M) = A \rightarrow B$ , where  $\text{Type}(\Gamma, x : A; N) \leq_{I(\Gamma)}^E M : B$  for some  $N \in \text{Gen}(M)$ :

From the induction hypothesis, we have  $\Gamma, x : A \vdash N : \text{Type}(\Gamma, x : A; N) \leq_{I(\Gamma)}^E M : B$ , and then  $\Gamma \vdash \lambda x : A.M : (A \rightarrow B) = \text{Type}(\Gamma, x : A; M)$ .

3.  $\text{Type}(\Gamma; M_1 M_2) = B_2$ , where  $\text{Type}(\Gamma; N_1) \leq^E M_1 : B_1 \rightarrow B_2$  and  $\text{Type}(\Gamma; N_2) \leq_{I(\Gamma)}^E M_2 : B_1$  for some  $N_i \in \text{Gen}(M_i)$ :

From the induction hypotheses, we have  $\Gamma \vdash N_1 : \text{Type}(\Gamma; N_1) \leq^E M_1 : B_1 \rightarrow B_2$  and  $\Gamma \vdash N_2 : \text{Type}(\Gamma; N_2) \leq_{I(\Gamma)}^E M_2 : B_1$ . Then  $\Gamma \vdash M_1 M_2 : B_2 = \text{Type}(\Gamma; M_1 M_2)$ .

The completeness is by induction on derivation.

- Case of  $\Gamma \vdash x : \Gamma(x)$  :

We always have  $\text{Type}(\Gamma; x) = \Gamma(x) \leq_{I(\Gamma)}^E \Gamma(x)$ .

- $\Gamma \vdash \lambda x : A.M : A \rightarrow B$  from  $\Gamma, x : A \vdash M : B$  :

From the induction hypothesis, we have  $\text{Type}(\Gamma, x : A; N) \leq_{I(\Gamma, A)}^E M : B$  for some  $N \in \text{Gen}(M)$ , and then  $\text{Type}(\Gamma; \lambda x : A.M) = A \rightarrow B$ .

- $\Gamma \vdash M_1 M_2 : B_2$  from  $\Gamma \vdash M_1 : B_1 \rightarrow B_2$  and  $\Gamma \vdash M_2 : B_1$  :

From the induction hypotheses, we have  $\text{Type}(\Gamma; N_1) \leq_{I(\Gamma)}^E M_1 : B_1 \rightarrow B_2$  and  $\text{Type}(\Gamma; N_2) \leq_{I(\Gamma)}^E M_2 : B_1$  for some  $N_i \in \text{Gen}(M_i)$ . Then we have  $\text{Type}(\Gamma; M_1 M_2) = B_2$ .

- $\Gamma \vdash M : \forall X.A$  from  $\Gamma \vdash M : A$  where  $X \notin \text{FV}(\Gamma)$ :

From the induction hypothesis, we have  $\text{Type}(\Gamma; N) \leq_{I(\Gamma)}^E M : A$  for some  $N \in \text{Gen}(M)$ , and then  $M : A \leq_{I(\Gamma)} M : \forall X.A$  since  $X \notin \text{FV}(\Gamma)$ . Hence, we have  $\text{Type}(\Gamma; N) \leq_{I(\Gamma)}^E M : \forall X.A$  for some  $N \in \text{Gen}(M)$ .

- $\Gamma \vdash M[X := B] : A[X := B]$  from  $\Gamma \vdash M : \forall X.A$ :

From the induction hypothesis, we have  $\text{Type}(\Gamma; N) \leq_{I(\Gamma)}^E M : \forall X.A$  for some  $N \in \text{Gen}(M)$ . Then we also have  $M : \forall X.A \leq^E M[X := B] : A[X := B]$ , and hence  $\text{Type}(\Gamma; N) \leq_{I(\Gamma)}^E M[X := B] : A[X := B]$  for some  $N \in \text{Gen}(M)$  from the transitivity.  $\square$

## References

- [1] H. P. Barendregt: *The lambda Calculus. Its Syntax and Semantics*, North-Holland, second, revised edition, 1984.
- [2] H. P. Barendregt: *Lambda calculi with types*, In S. Abramsky, et al. editors, *Handbook of Logic in Computer Science*, Vol II, pp. 117–309, Oxford University Press, 1992.
- [3] H. P. Barendregt, W. Dekkers, R. Statman: *Lambda Calculus with Types*, Cambridge University Press, 2012.
- [4] G. Barthe, M. H. Sørensen: *Domain-Free Pure Type Systems*, Lecture Notes in Computer Science 1234, pp. 9–20, 1997.
- [5] G. Dowek: *Higher-Order Unification and Matching*, In A. Robinson and A. Voronkov editors, *HANDBOOK OF AUTOMATED REASONING*, Elsevier Science Publishers, pp. 1009–1062, 2001.

- [6] K. Fujita, A. Schubert: *Partially typed terms between Church-style and Curry-style*, Lecture Notes in Computer Science 1872, pp. 505-520, 2000.
- [7] K. Fujita, A. Schubert: *The undecidability of type related problems in type-free style System F*, Leibniz International Proceedings in Informatics 6, pp. 103-118, 2010.
- [8] K. Fujita: ラムダ計算の型問題について, submitted.
- [9] M. Odersky, K. Läufer: *Putting Type Annotations to Work*, POPL '96, pp. 54-67, 1996.
- [10] A. Schubert: *Second-order unification and type inference for Church-style polymorphism*, POPL '98: Proc. 25th ACM Symposium on Principles of Programming Languages, pp. 279-288, 1998.
- [11] J.B. Wells: *Typability and type checking in system F are equivalent and undecidable*, Ann. Pure Appl. Logic 98, pp. 111-156, 1999.